Detection of VPN-Based Fraud Using Network Analysis

Aryan Rajeshkumar

BSc (Hons.) Computer Science
Honours Dissertation

Supervised by Prof. Abrar Ullah

Heriot-Watt University

School of Mathematical and Computer Sciences

March 2025

DECLARATION

I, Aryan Rajeshkumar, confirm that this work submitted for assessment is my own and is ex-pressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: Aryan

Date: 03/08/2025

ABSTRACT

This study focuses on improving fraud detection carried out using Virtual Private Networks (VPN) to bypass location-based authentication. This way fraudsters can anonymously conduct criminal activity which would theoretically be non-detectable. By analyzing network features VPNs cause, this research aims to develop and compare AI models for identifying suspicious activity. The proposed models aim to enhance fraud detection by classifying abnormal network patterns, thereby improving the overall security of financial systems. This study will employ the pre-defined dataset ISCXVPN2016 created by University of New Brunswick [www.unb.ca, n.d., 2016] that compares VPN and Non-VPN traffic over multiple protocols. The goal is to implement a comprehensive suite of Artificial Intelligence (AI) models including Convolutional Neural Networks, Decision Trees, Gradient Boosted Trees, K-Nearest Neighbors, Logistic Regression, Multilayer Perceptron, Naive Bayes, Random Forest, Recurrent Neural Network, and Support Vector Machines on this dataset. These models will then be compared using performance metrics such as accuracy, precision, recall, F1-score, and ROC curves.

The highest accuracies were achieved by ensemble-based methods, specifically Gradient Boosted Trees (92.60%) and Random Forest (92.47%), highlighting their importance in classifying encrypted traffic. In contrast, simpler models such as Logistic Regression (64.11%) and Naïve Bayes (54.23%) underperformed due to their inability to capture complex non-linear patterns. These findings will help in creating a real-time fraud detection solution in modern financial systems.

Keywords: VPN detection, Network analysis, Fraud detection, Machine Learning.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Prof. Abrar Ullah (Supervisor) for this guidance and support for this dissertation.
I'm also grateful to Heriot Watt University Dubai for providing the necessary resources for this research.
Special thanks to my family for their support throughout this journey. Their encouragement kept me motivated throughout the challenging phases of this dissertation.
I would like to acknowledge the effort of everyone who has contributed directly or indirectly to this research.

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

Point of Sale (POS) machines and Payment Gateways are an integral part of modern financial systems. With increasing adoption of Internet of Things (IoT) technologies, POS machines now operate via Wi-Fi, connecting to the internet to process payments in real time. Payment Gateways serves as a bridge between the online merchant and the financial institution they're associated with. However, the expansion of such systems has introduced new vulnerabilities, particularly related to network-based fraud. One such vulnerability is using Virtual Private Networks (VPNs). Malicious Actors could use VPNs to bypass geolocation-based restrictions and commit fraudulent transactions. This report explores a novel approach to mitigate VPN-based fraud by analyzing changes in network features, which is a common characteristic of VPN connections, as an indicator of suspicious activities.

This chapter presents the motivation, aim, and contributions of the research, outlining the importance of the topic and its relevance to the financial sector. Additionally, we will discuss the limitations of this study and identify the target audience.

1.1 Motivation

As digital payments and paperless transactions have become increasingly important, financial institutions rely heavily on geolocation data as a key part in authentication and compliance with regulatory bodies, such as country-specific sanctions and pricing. Fraudsters are now exploiting VPN technology to obscure their true location and/or residence, thereby bypassing crucial safeguards. This exposes financial institutions and merchants to non-compliance with rules set for them, leading to fines being issued and in general loss of customer confidence.

1.2 Aim

This research aims to develop, implement and compare different Artificial Intelligence (AI) models, such as K-Nearest Neighbors (KNN) and Gradient Boosted Trees (GBT), for detecting VPN based network traffic over multiple protocols like User Datagram Protocol (UDP), Transfer Control Protocol (TCP) and Hypertext Transfer Protocol Secure (HTTPS). Using network analysis, this report seeks to determine which model most effectively identifies VPN based network traffic.

1.3 Objectives

- **Literature Review and Requirement Analysis**: Conducting a comprehensive review and analysis of current VPN detection techniques and to explore upon the gaps in existing solutions. This will be completed within the first part of the project.
- **Data Collection**: Gain access to VPN-nonVPN dataset (ISCXVPN2016), which contains network traffic over multiple different protocols.
- **Methodology**: Develop a detailed methodology that applies network analysis to analyze network patterns to detect VPN traffic.
- **AI Model Implementation**: Implementation of multiple AI models (such as KNN and GBT) and conducting hyperparameter optimization for best performance.
- **Performance Analysis**: Evaluating each of the proposed model's performance using metrics such as accuracy, precision, recall, F1 score.
- **Comparative Evaluation**: Conducting comparative analysis to proposed AI models with existing research and fraud detection methods to identify improvements and operational effectiveness in real world scenarios.

1.4 Organization

The next sections will build upon VPN encrypted traffic classification. Section 2 contains all background and literature reviews pertaining to VPN detection and fraud detection. Section 3 contains the methodology behind building ML models to classify VPN traffic. Section 4 contains all the results achieved after creating the models. Section 5 will analyze the results achieved and explain their reasons. Section 6 and 7 will discuss and conclude the findings and the study.

2 BACKGROUND

2.1 Introduction

The rapid expansion of digital payment systems, both POS and online payments, has transformed how transactions are processed. However, this change has led to the creation of multiple sophisticated fraudulent schemes. One such scheme is when the attacker masks their true location to bypass any authentication control set up by the financial (or Authoritative) body. This chapter will review the literature available on fraud and VPN detection, including advances in pattern discovery, ML, network analysis and other hybrid models. The chapter will end with a critical analysis of the literature and summary.

2.2 Evolution of Payment Systems and their Emerging Vulnerabilities

Modern payment systems have evolved from traditional POS devices to dynamic IoT solutions and online processing. This has increased its reliance on wireless connectivity and cloud-based processing and has drastically improved convenience and transaction efficiency. This has inadvertently introduced new vulnerabilities. For example, many IoT-based POS terminals lack built-in GPS, making them susceptible to location spoofing. Another example is bypassing YouTube costing where the fraudulent actor may spoof their location to get cheaper costing for purchases and subscription [Cybernews, 2022].
Moreover, it has been demonstrated that vulnerabilities in payment systems are increased by the ease with which cybercriminals demonstrate and distribute sophisticated fraud methods online. Such online tutorials on credit card fraud showcased the entire process, including the usage of VPNs and SOCKS5 proxies to purchase stolen credit card details and cashing them out [van Hardeveld, Webber and O'Hara, 2016].
Such tactics not only undermine regulatory compliance but also hurt customer trust in digital payment ecosystems and merchants.

2.3 Approaches to Fraud Detection: Pattern Discovery and Early Techniques

Early fraud detection techniques focused on pattern discovery within transactions datasets. The use of transaction logs (TLogs) in combination with video data to identify irregular cashier activities using the Teiresias algorithm have proven to be a viable solution for fraud detection over a local scale [Gabbur et al. 2011]. These methods helped identify behavioral inconsistencies [Kelly 2019] but were often limited by scalability and could not fully capture complex fraud patterns. This laid out the groundwork for future more sophisticated techniques.

2.4 Machine Learning and AI applied for Fraud Detection

Machine learning has revolutionized fraud detection by allowing for the analysis of larger and more complex datasets. Decision tree algorithms and Support Vector Machines (SVMs) have been compared for effectiveness for fraud detection, noting that decision trees performed better over highly imbalanced datasets, which is a common feature of fraud detection datasets [Sahin and Duman 2011]. More recent studies have employed probabilistic scoring methods to prioritize higher risk transactions [Chugh, Malik, Gupta, and Alkahtani, 2025]. Despite these solutions, the challenge remains in handling data imbalances and ensuring the model is both accurate and efficient. This dissertation extends this line of research by comparing various AI models (like KNN and GBT) using network analysis to determine the most effective solution for detecting VPN based fraud.

Other studies have shown effective use of neural and probabilistic in fraud detection within communication networks. This involves combining feed-forward neural networks, Gaussian mixture models, and Bayesian networks to identify fraudulent behavior based on toll ticket data [Taniguchi et al. 2002].

Logistic Regression, Random Forest, Naïve Bayes, and Multilayer Perceptron were evaluated on highly imbalanced credit card fraud datasets. Implementation of oversampling via SMOTE and feature selection to overcome data imbalance has been crucial to achieve high classification performance. Most notably, Random Forest classifier achieved a test accuracy of 99.96% on Credit Card Fraud Detection dataset showing the strength of an ensemble model for non-linear imbalanced datasets [Varmedja et al., 2019].

Comparative analyses further revealed that instance-based methods like KNN can achieve high accuracies and low false positives when combined with hybrid sampling techniques. KNN outperformed both naïve Bayes and logistic regression on highly skewed credit card fraud datasets [Awoyemi, Adetunmbi and Oluwadare, 2017].

A sliding window strategy was employed to cluster cardholders based on their transaction amounts, allowing for extraction of dynamic behavioral features that adapt to evolving fraud patterns. This method mitigates the impact of imbalanced datasets [Dornadula and Geetha, 2019].

2.5 Network Analysis and VPN Detection

Network analysis offers a powerful approach to detecting VPN usage, which is being used by fraudsters to spoof geo-location data to bypass security measures. Techniques such as analyzing packet lengths, Domain Name System (DNS) queries, and flow durations have been explored to distinguish between VPN and non-VPN traffic. Non-standard HTTPS ports and other identifiable server attributes can also serve as indicators of potential VPN usage [Abideen, Saleem, and Ejaz 2019].

While ML models like Multilayer Perceptron (MLP) networks have been applied and show promising results [Hines and Youssef 2018], there is a need for a systematic comparison of multiple AI models in this context. Random Forest Classifiers, combined with feature

reduction, hyperparameter optimization and overfitting prevention demonstrated high efficiency achieving ROC values between 95-98.6% across different timeouts [Al-Fayoumi, Al-Fawa'reh and Nashwan, 2022].

Table. 1. Random Forest experimental results for VPN detection

Timeout	Class label	Feature selection	No. of features	Precession	Recall	F1	ROC area	Accuracy
15	Non-VPN	N	24	0.954	0.97	0.962	0.986	95.02%
	VPN		24	0.942	0.914	0.928	0.986	
	Non-VPN	Y	9	0.907	0.949	0.927	0.96	90.33%
	VPN		9	0.896	0.819	0.856	0.96	
	Non-VPN	GA	8	0.927	0.907	0.917	0.976	92.06%
	VPN		8	0.915	0.933	0.924	0.976	
30	Non-VPN	N	24	0.946	0.966	0.956	0.98	94.12%
	VPN		24	0.932	0.893	0.912	0.98	
	Non-VPN	Y	9	0.905	0.946	0.925	0.95	89.92%
	VPN		9	0.886	0.809	0.846	0.95	
	Non-VPN	GA	8	0.939	0.956	0.947	0.976	93.01%
	VPN		8	0.912	0.88	0.896	0.976	
60	Non-VPN	N	24	0.927	0.891	0.908	0.978	93.06%
	VPN		24	0.933	0.956	0.944	0.978	
	Non-VPN	Y	9	0.881	0.832	0.856	0.952	89.14%
	VPN		9	0.898	0.929	0.913	0.952	
	Non-VPN	GA	8	0.908	0.876	0.892	0.974	91.76%
	VPN		8	0.923	0.944	0.934	0.974	
120	Non-VPN	N	24	0.92	0.891	0.906	0.978	93.50%
	VPN		24	0.943	0.958	0.95	0.978	
	Non-VPN	Y	9	0.854	0.817	0.835	0.943	88.71%
	VPN		9	0.904	0.925	0.914	0.943	
	Non-VPN	GA	8	0.91	0.87	0.89	0.974	92.47%
	VPN		8	0.932	0.954	0.943	0.974	

Ensemble models like GBT and Random Forest have shown success with 96.7% and 97.6% respectively in real-Time VPN anomaly detection system [Alamleh et al., 2024]. This is also seen for binary classification of VPN traffic using wavelet features where Random Forest was superior compared to neural networks [Sajid and Pekar, 2025]. Usage of models like Random Forest and MLP to classify VPN and non-VPN network traffic can overcome the limitations of rule-based systems by being able to capture complex network patterns [Goel et al., 2022].

2.6 Hybrid Models in Fraud Detection

Hybrid models combine strengths of ML and network analysis to improve fraud detection and overcome the limitations of the standalone approaches. The Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) hybrid network leverages the special feature extraction capabilities of CNNs and temporal pattern recognition of RNNs. This combined approach can effectively capture both structured patterns in transactional data

and sequential dependencies in user behavior, resulting in high detection accuracy, low false positives, and faster efficiencies which are essential for real time fraud detection [Lenka and Tiwari, 2020].
A stacked RNN model to detect fraudulent transactions have also shown success. The ability of deep recurrent architectures to adapt to evolving transaction patterns can achieve high accuracies with minimal error rates. [Dutta and Bandyopadhyay 2020].

2.7 Critical Analysis

While the literature presents a wide range of methodologies for fraud and VPN detection, several critical gaps in the studies remain.

- **Integration of Network Analysis**: Many studies focus separately on ML or network analysis. Only few provide a holistic approach that integrates both, particularly in context of VPN based fraud detection [van Hardeveld, Webber and O'Hara, 2016; Gabbur et al. 2011; Kelly 2019].
- **Comparative Evaluation of AI Models**: There is a lack of comprehensive studies evaluating a large variety of AI algorithms within the context of VPN based fraud detection [Sahin and Duman 2011; Chugh, Malik, Gupta, and Alkahtani, 2025; Taniguchi et al. 2002; Varmedja et al., 2019; Awoyemi, Adetunmbi and Oluwadare, 2017; Dornadula and Geetha, 2019].
- **Scalability and Real-Time Applicability**: Several of the approaches show a high accuracy and low false positives in controlled environments but fail to address scalability and computational efficiency in real-world scenarios, where there will be significantly higher volumes of transactions, hardware limitations and a need for real-time VPN detection [Abideen, Saleem, and Ejaz 2019; Hines and Youssef 2018; Al-Fayoumi, Al-Fawa'reh and Nashwan, 2022; Alamleh et al., 2024; Sajid and Pekar, 2025; Goel et al., 2022].
- **Regulatory and Ethical Considerations**: Current research often overlooks the implications of deploying an advanced fraud detection system in practice, some of which might be intrusive. Considering regulatory compliance and ethical challenges is crucial.
- **Outdated References and Techniques**: Some of the foundational papers have become dated, stressing the need for updated methodologies that consider the latest fraud detection strategies and technological advancements [Gabbur et al. 2011; van Hardeveld, Webber and O'Hara, 2016; Taniguchi et al. 2002].

2.8 Summary

Overall, the literature shows a fast-evolving landscape in digital payments and fraud detection. Techniques such as location spoofing and VPN-based masking enable fraudsters to bypass security measures and regulatory controls. Early fraud detection techniques depended upon pattern recognition with transaction logs. This has given way to the creation of much more sophisticated techniques using multiple different ML models and hybrid models. Despite these advancements, the challenge remains, particularly in VPN based fraud detection using network analysis which this review will underscore.

In the next section, Section 3, we detail the methodology behind VPN detection.

3 METHOD

3.1 Introduction

This chapter outlines the approach adopted to detect VPN-based fraud using network analysis. This will involve:

- **Data Acquisition**: This study uses the ISCXVPN2016 dataset [www.unb.ca, n.d., 2016] which is a publicly available dataset which contains labeled network traffic for both VPN and non-VPN scenarios.
- **Preprocessing and Feature Selection:** The raw data cannot be used directly for model training since the datasets for VPN and non-VPN are different. After preprocessing, there is a need for extraction of time-related and network specific features.
- **Environment Setup**: This section will discuss the necessary tools required to recreate the experiment conducted in this study.
- **Model Implementation**: This section includes all the models tested in this study and their method.
- **Model Evaluation**: This section explains the methodology used for evaluating the performance of the models.

3.2 Dataset

The ISCXVPN2016 dataset provides a diverse collection of network traffic data. The dataset was designed to simulate real-world scenarios and contains both VPN and non-VPN network traffic across 14 different protocols (e.g., VOIP, P2P, SMTP, etc.). The key features of the dataset include:

- **Traffic Capture**: All data was collected using packet capturing/sniffing tools like Wireshark and tcpdump.
- **Session Types**: The traffic captured includes both VPN and non-VPN scenarios and has been differentiated.

- **Traffic Categories**: The premade datasets include detailed features for each capture which aids in the proper labeling of data for supervised learning.
- **Raw Files**: The original pcap files have also been provided to replicate the actual dataset creation steps.
- **Time Based Data**: Datasets have been provided over 4 timescales (15s, 30s, 60s and 120s). Each timescale has 2 datasets, VPN and non-VPN.

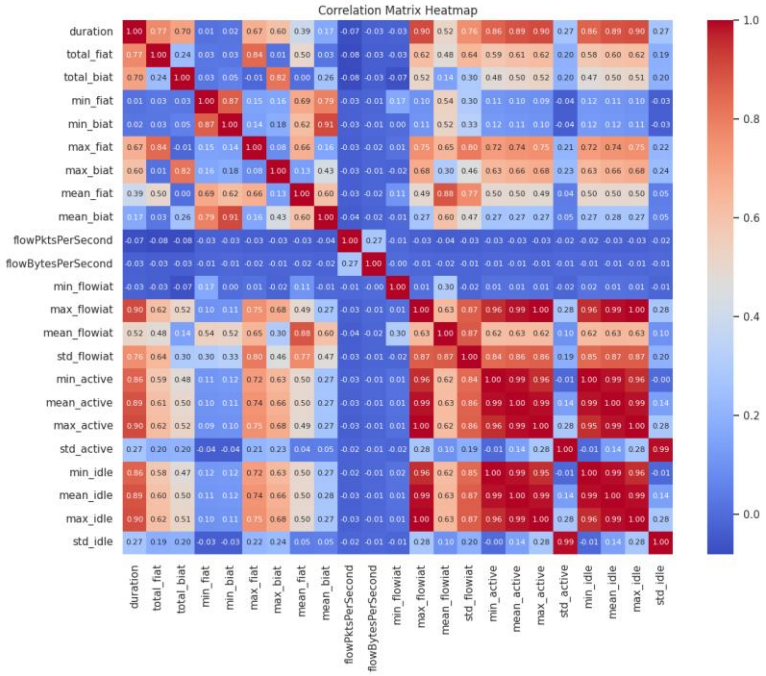

Fig. 1. Correlation Matrix of dataset attributes

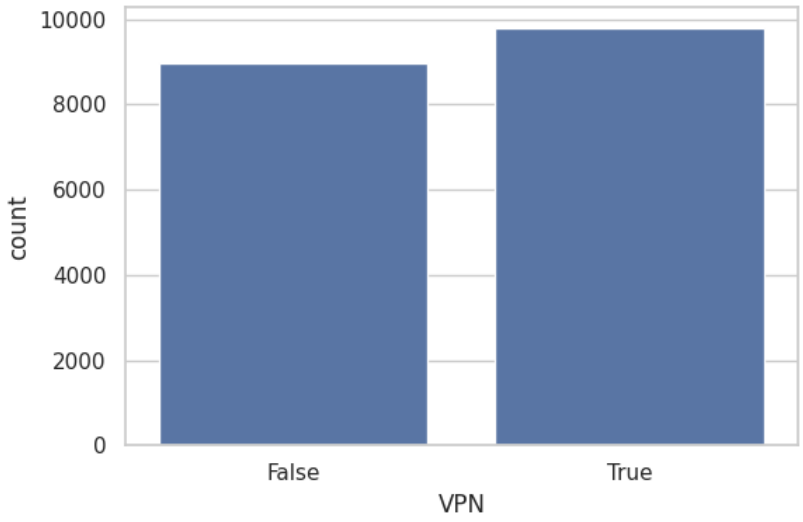

Fig. 2. Class distribution of dataset

Table. 2. Dataset columns and types

No.	Column	Non-Null Count	Data type
0	duration	18758 non-null	int64
1	total_fiat	18758 non-null	int64
2	total_biat	18758 non-null	int64
3	min_fiat	18758 non-null	int64
4	min_biat	18758 non-null	int64
5	max_fiat	18758 non-null	int64
6	max_biat	18758 non-null	int64
7	mean_fiat	18758 non-null	float64

8	mean_biat	18758 non-null	float64
9	flowPktsPerSecond	18758 non-null	float64
10	flowBytesPerSecond	18758 non-null	float64
11	min_flowiat	18758 non-null	int64
12	max_flowiat	18758 non-null	int64
13	mean_flowiat	18758 non-null	float64
14	std_flowiat	18758 non-null	float64
15	min_active	18758 non-null	int64
16	mean_active	18758 non-null	float64
17	max_active	18758 non-null	int64
18	std_active	18758 non-null	float64
19	min_idle	18758 non-null	int64
20	mean_idle	18758 non-null	float64
21	max_idle	18758 non-null	int64
22	std_idle	18758 non-null	float64
23	class1	18758 non-null	object
24	VPN	18758 non-null	bool

3.3 Preprocessing and Feature Selection

Due to the raw nature of the datasets provided, preprocessing is necessary to ensure the quality of the input for the ML models.

The following steps were taken to ensure the dataset would be ideal for training.

- **Data generation**

- Labeling: A column has been added to each dataset called “VPN” and contains values “TRUE” or “FALSE” depending on if the dataset was VPN or non-VPN.
- Merging: The VPN and non-VPN datasets for each timescale have been merged and shuffled to create 4 final labeled datasets.

- **Data Cleaning**
  - Removal of Redundancies: Removing duplicate records and empty entries.
  - Consistency Check: Checking for bogus data or mismatched types.
- **Data Normalization**
  - Scaling: Normalizing numerical values to reduce impact of extreme values.
- **Feature Selection**
  - Key time-related features are extracted to characterize network flow [Bagui et al., 2017].

Table. 3. Dataset features

Feature	Description
Duration	The duration of the flow
Fiat	Forward Inter Arrival Time, the time between two packets sent forward direction (mean, min, max, std)
Biat	Backward Inter Arrival Time, the time between two packets sent backwards (mean, min, max, std)
Flowiat	Flow Inter Arrival Time, the time between two packets sent in either direction (mean, min, max, std)
Active	The amount of time, the time a flow was active before going idle (mean, min, max, std)
Idle	The amount of time, the time a flow was idle before becoming active (mean, min, max, std)
fb_psec	Flow bytes per second
fp_psec	Flow packets per second
Timeout	Flow timeout

3.4 Environment Setup

- Operating System: Ubuntu (Using Windows Subsystem for Linux).
- Code editor: Visual Studio code (using Jupyter Notebook).
- Packages used
  - Pandas: For data handling.
  - Matplotlib: For plotting graphs.
  - Scikit-learn: For model training.

- TensorFlow: For model training (CNN and RNN).

3.5 Model Implementation

This research focuses on developing and optimizing multiple ML models using supervised learning on the VPN non-VPN dataset. The implementation details are as follows:

- **Model Selection**
  - K-Nearest Neighbors (KNN)
  - Logistic Regression (LR)
  - Support Vector Machine (SVM)
  - Naïve Bayes (NB)
  - Random Forest (RF)
  - Gradient Boosting Trees (GBT)
  - Decision Trees (DT)
  - Multilayer Perceptron (MLP)
  - Recurrent Neural Network (RNN)
  - Convolutional Neural Network (CNN)

These models are chosen based on their established performance in classification [Bagui et al., 2017].

- **Training and Validation**
  - Data Splitting: Splitting the data into training, validation and test sets using stratified sampling to maintain the class distribution.
  - Cross-Validation: k-fold cross validation is employed to ensure better model performance.
  - Hyperparameter Tuning: Grid search and randomized search strategies are used to optimize model parameters.
- **Model Training**
  - Feature Selection: Recursive feature elimination (RFE) is used to select the most relevant features from the dataset.
  - Algorithm Implementation: The supervised ML models are implemented using scikit-learn.

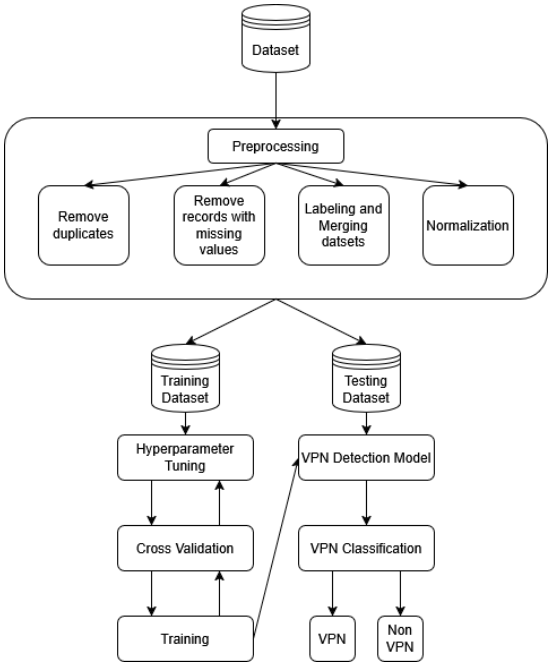

Fig. 3. Training-Testing Workflow

3.6 Model Evaluation

To assess and compare the models, multiple metrics are used. These metrics are:

- Accuracy: The overall percentage of correctly classified instances.

$$Accuracy = \frac{TP + TN}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{3.1}$$

  Where:
  - TP = True Positives
  - TN = True Negatives
  - FP = False Positives
  - FN = False Negatives
- Precision: Proportion of positive predictions that are actually correct.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3.2}$$

- Recall (Sensitivity): Proportion of actual positives that are correctly identified.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

- F1 Score: The mean of precision and recall, balancing between the two.

$$F1\ Score = 2\ \times \frac{\text{Precision}\ \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

- ROC Curve: Plots Recall against False Positive Rate (FPR) for different threshold values.

$$FPR = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{3.5}$$

- AUC: This is the area under the ROC curve. Shows the overall ability to classify between the positive and negative classes.

$$AUC = \int_0^1 TPR(FPR)\ d(FPR) \tag{3.6}$$

  Where:
  - TPR = True Positive Rate

3.7 Summary

Multiple supervised learning models are implemented using the dataset which has been preprocessed by labeling, merging, cleaning and normalization. These models be evaluated using performance metrics which are industry standard.

In the next chapter, Section 4, we present the findings of our research.

4 RESULTS

This chapter presents the quantitative evaluation and qualitative interpretation of the models developed to detect VPN-based network traffic. For each model, test accuracy, classification metrics (precision, recall, F1-score), and include the confusion matrix and ROC curve are included. For models that underwent hyperparameter tuning, specifically those tuning learning rate and maximum depth (such as GBT), a plot of cross validation accuracy versus learning rate for different maximum depths is provided. The results are summarized in Table 3 and discussed in the following sections.

4.1 Experiment 1: K-Nearest Neighbors (KNN)

- Setup and Parameters:
    - Algorithm: KNN
    - Parameters Tested: Number of k neighbors [3, 5, 7, 9, 11].
    - Best Parameter: k = 3
- Test Accuracy: 83.05%
- Classification Report:
    - Class 0: Precision 0.82, Recall 0.84, F1-Score 0.83
    - Class 1: Precision 0.84, Recall 0.83, F1-Score 0.83
- Evaluation Graphs

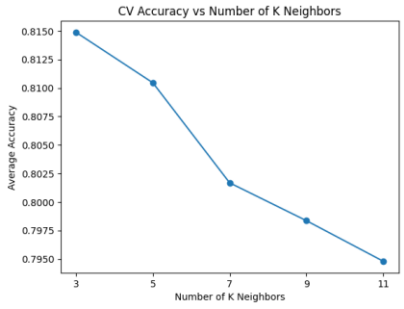

Fig. 4. KNN Cross-Validation

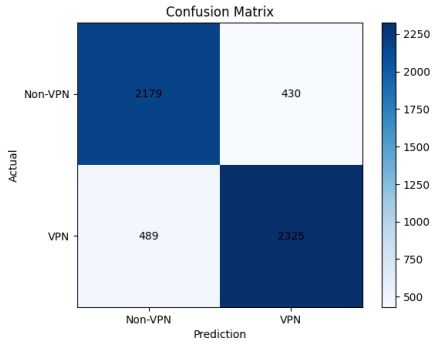

Fig. 5. KNN Confusion Matrix

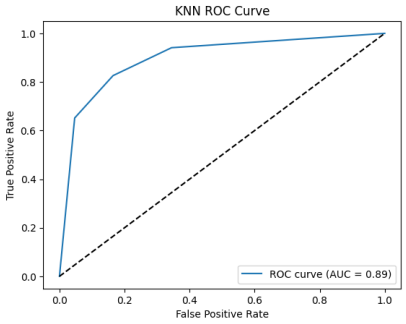

Fig. 6. KNN ROC Curve

4.2 Experiment 2: Logistic Regression (LR)

- Setup and Parameters:
  - Algorithm: LR
  - Parameters Tested: Regularization strength C [0.01, 0.1, 1, 10, 100].
  - Best Parameter: C = 100
- Test Accuracy: 64.11%
- Classification Report:
  - Class 0: Precision 0.67, Recall 0.50, F1-Score 0.57
  - Class 1: Precision 0.62, Recall 0.77, F1-Score 0.69

4.3 Experiment 3: Support Vector Machines (SVM)

- Setup and Parameters:
  - Algorithm: SVM
  - Parameters Tested

        - Penalty Parameter C [0.1, 1, 10, 100]
        - Gamma value for Radial Basis Function (RBF) kernel [0.001, 0.01, 0.1, 1]
    - Best Parameters
        - C = 100
        - Gamma = 1
- Test Accuracy: 78.14%
- Classification Report:
    - Class 0: Precision 0.74, Recall 0.85, F1-Score 0.79
    - Class 1: Precision 0.84, Recall 0.71, F1-Score 0.77
- Evaluation Graphs

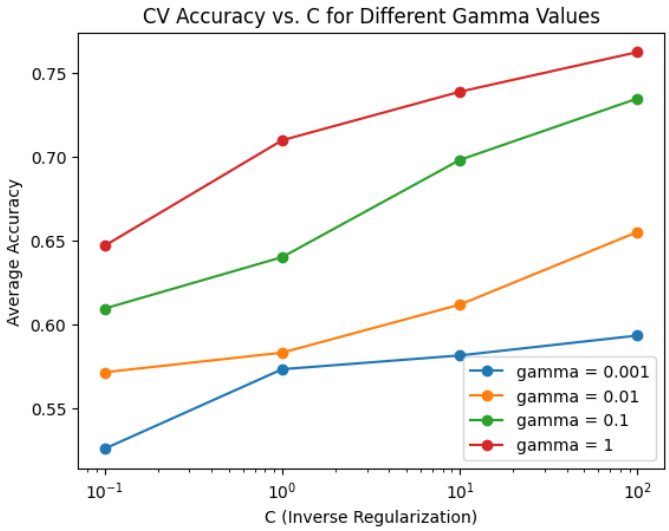

Fig. 7. SVM Cross-Validation

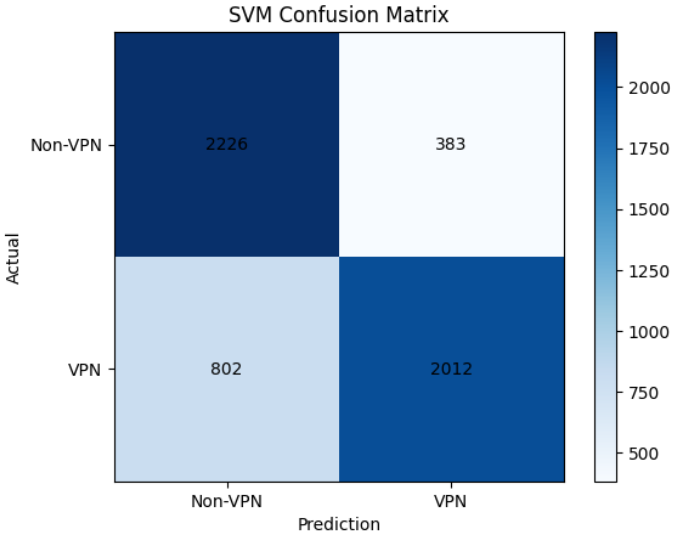

Fig. 8. SVM Confusion Matrix

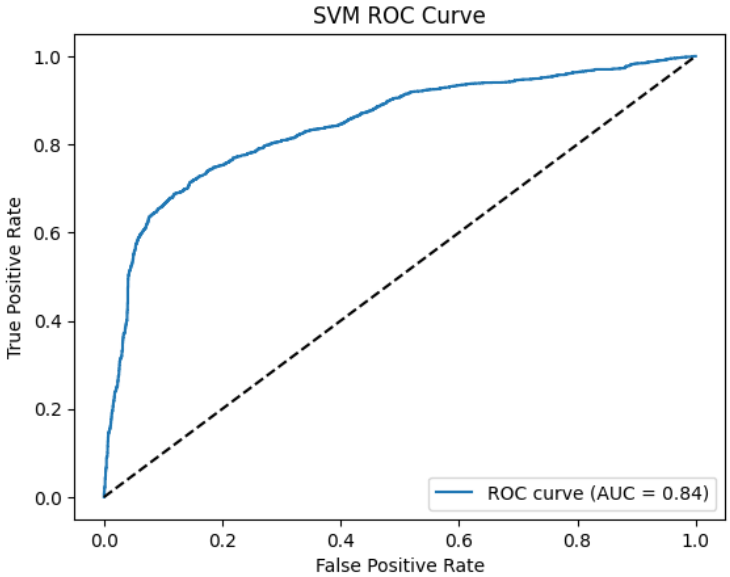

Fig. 9. SVM ROC Curve

4.4 Experiment 4: Naïve Bayes (NB)

- Setup and Parameters:
    - Algorithm: NB
    - Parameters Tested: Variance smoothing parameter [1e-9, 1e-8, 1e-7, 1e-6, 1e-5].
    - Best Parameter: var_smoothing = 1e-09
- Test Accuracy: 54.23%
- Classification Report:
    - Class 0: Precision 0.82, Recall 0.06, F1-Score 0.12
    - Class 1: Precision 0.53, Recall 0.99, F1-Score 0.69

4.5 Experiment 5: Random Forest (RF)

- Setup and Parameters:
    - Algorithm: RF
    - Parameters Tested
        - Number of estimators [50, 100, 200].
        - Maximum depth [None, 10, 20].
        - Minimum samples split [2, 5].
        - Minimum samples leaf [1, 2].
    - Best Parameters
        - n_estimators = 100
        - max_depth = 20
        - min_samples_split = 2
        - min_samples_leaf = 2
- Test Accuracy: 92.47%
- Classification Report:
    - Class 0: Precision 0.93, Recall 0.91, F1-Score 0.92
    - Class 1: Precision 0.92, Recall 0.94, F1-Score 0.93
- Evaluation Graphs

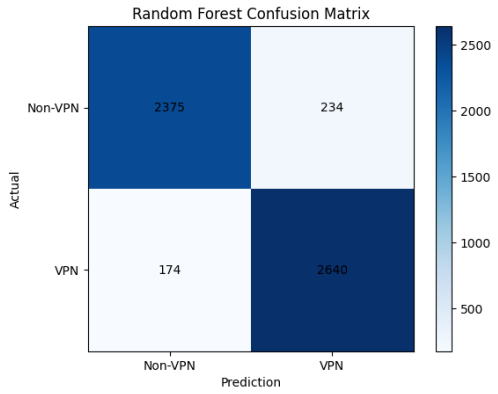

Fig. 10. RF Confusion Matrix

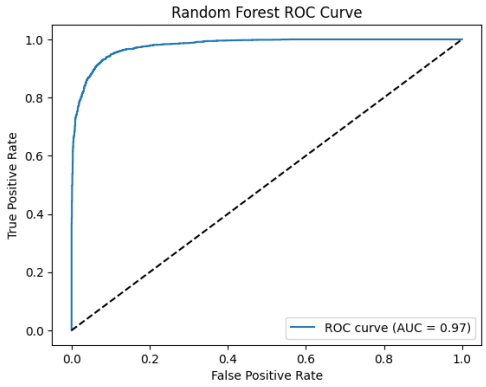

Fig. 11. RF ROC Curve

4.6 Experiment 6: Gradient Boosted Trees (GBT)

- Setup and Parameters:
  - Algorithm: GBT
  - Parameters Tested
    - Number of estimators [50, 100, 200]
    - Learning rate [0.01, 0.1, 0.2]
    - Maximum depth [3, 5, 7]
  - Best Parameters
    - n_estimators = 200
    - learning_rate = 0.2
    - max_depth = 7

- Test Accuracy: 92.60%
- Classification Report:
    - Class 0: Precision 0.92, Recall 0.92, F1-Score 0.92
    - Class 1: Precision 0.93, Recall 0.93, F1-Score 0.93
- Evaluation Graphs

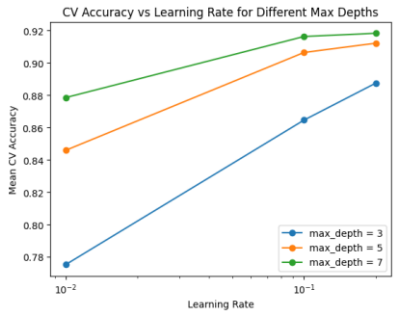

Fig. 12. GBT Cross-Validation

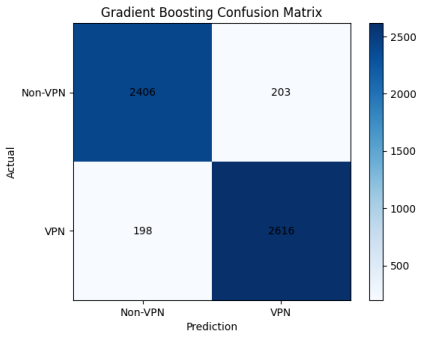

Fig. 13. GBT Confusion Matrix

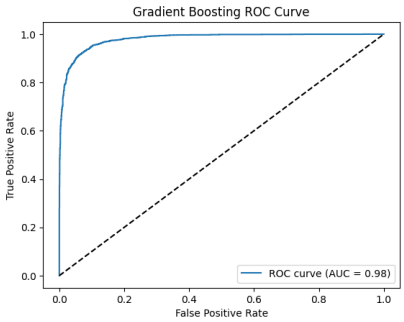

Fig. 14. GBT ROC Curve

4.7 Experiment 7: Decision Trees (DT)

- Setup and Parameters:
    - Algorithm: DT
    - Parameters Tested
        - Maximum depth [None, 5, 10, 20]
        - Minimum samples split [2, 5, 10]
        - Minimum samples leaf [1, 2, 4]
    - Best Parameters
        - max_depth = 20
        - min_samples_split = 2
        - min_samples_leaf = 1
- Test Accuracy: 89.39%
- Classification Report:
    - Class 0: Precision 0.89, Recall 0.89, F1-Score 0.89
    - Class 1: Precision 0.90, Recall 0.89, F1-Score 0.90
- Evaluation Graphs

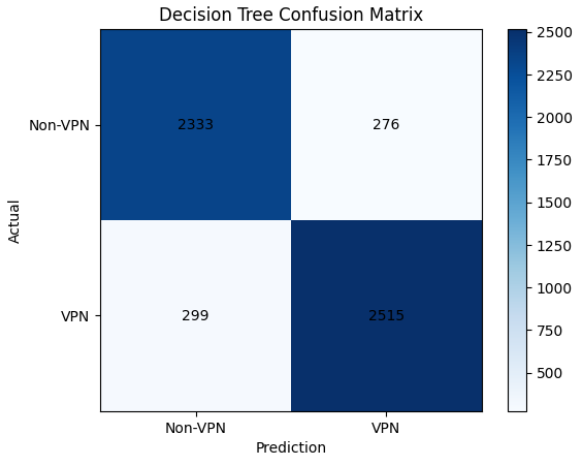

Fig. 15. DT Confusion Matrix

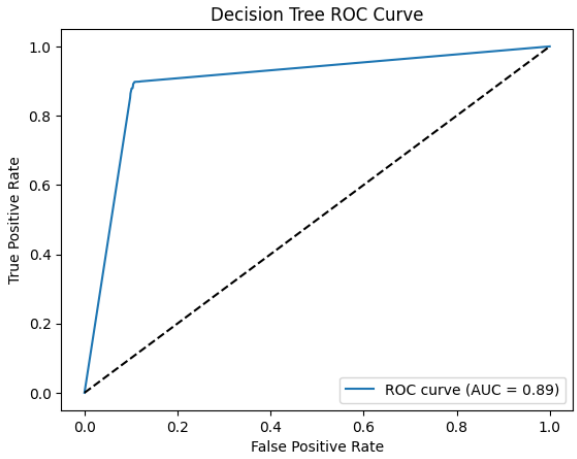

Fig. 16. DT ROC Curve

4.8 Experiment 8: Multilayer Perceptron (MLP)

- Setup and Parameters:
  - Algorithm: MLP
  - Parameters Tested
    - Hidden layer sizes [(50,), (100,), (50,50), (100,50)]
    - Activation ['relu', 'tanh']
    - Solver [‘adam’]
    - Alpha [0.0001, 0.001]
    - Learning rate ['constant', 'adaptive']
  - Best Parameters
    - hidden_layer_sizes = (100, 50)
    - activation = ‘relu’
    - solver = ‘adam’
    - alpha = 0.0001
    - learning_rate = ‘constant’
- Test Accuracy: 72.98%
- Classification Report:
  - Class 0: Precision 0.70, Recall 0.77, F1-Score 0.73

- Class 1: Precision 0.77, Recall 0.69, F1-Score 0.73

- Evaluation Graphs

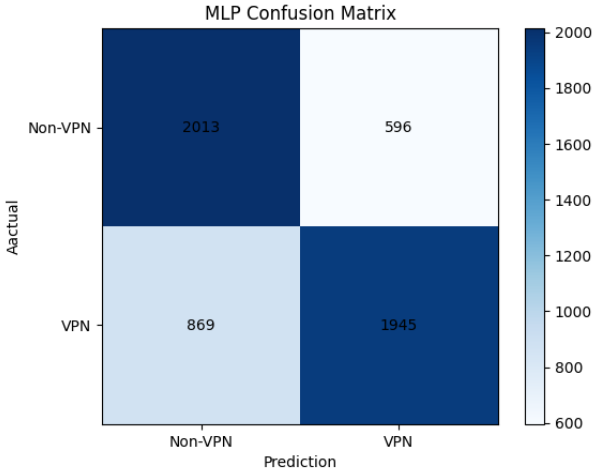

Fig. 17. MLP Confusion Matrix

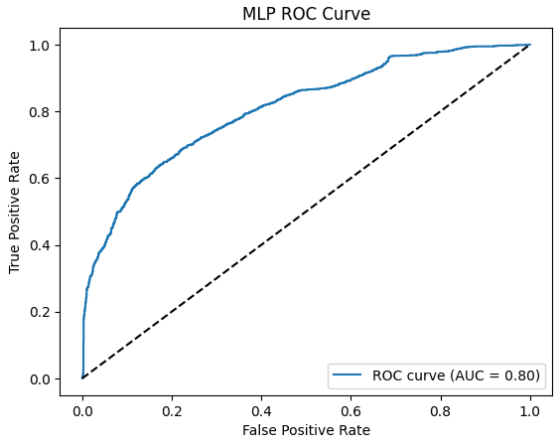

Fig. 18. MLP ROC Curve

4.9 Experiment 9: Recurrent Neural Networks (RNN)

- Algorithm: RNN
- Setup and Parameters:
  - Model Architecture:

        - SimpleRNN layer: 32 units with tanh activation (1792 parameters).
        - Dense layer: 16 neurons with ReLU activation (528 parameters).
        - Output dense layer: 1 neuron with sigmoid activation (17 parameters).
    - Model compilation:
        - Optimizer = 'adam'
        - Loss function = 'binary_crossentropy'
        - Metrics = 'accuracy'
    - Training configuration:
        - Epochs = 200
        - Batch size = 32
        - Validation split = 0.2 (20%)
- Test Accuracy: 74.34%
- Classification Report:
    - Class 0: Precision 0.71, Recall 0.79, F1-Score 0.75
    - Class 1: Precision 0.78, Recall 0.70, F1-Score 0.74
- Evaluation Graphs

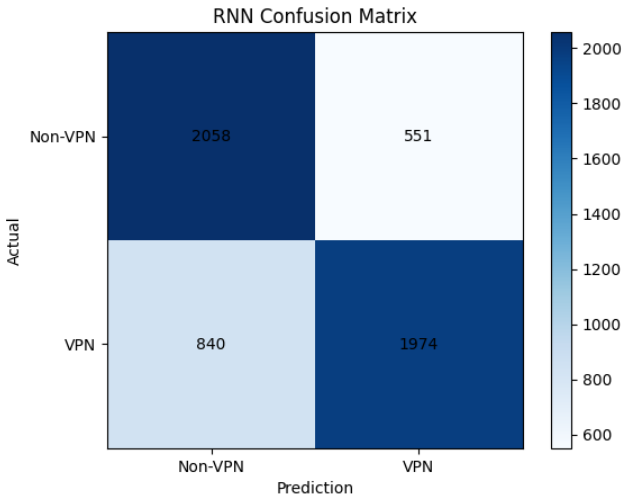

Fig. 19. RNN Confusion Matrix

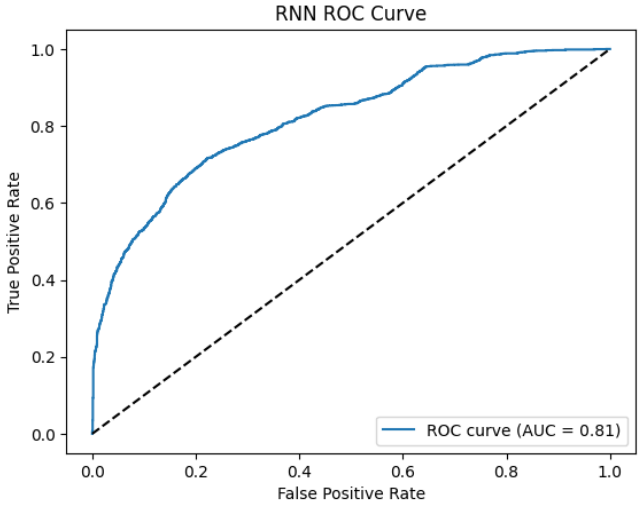

Fig. 20. RNN ROC Curve

4.10 Experiment 10: Convolutional Neural Networks (CNN)

- Algorithm: CNN
- Setup and Parameters:
    - Model Architecture:
        - First 1D convolutional layer:
            - Filters = 32
            - Kernel size = 3
            - Parameters = 128
        - First MaxPooling1D layer Pool size = 2
        - Second 1D convolutional layer:
            - Filters = 64
            - Kernel size = 3
            - Activation = ReLU
            - Parameters 6208
        - Second MaxPooling1D layer Pool size = 2
    - Model compilation:
        - Optimizer = ‘adam’

        - Loss function = ‘binary_crossentropy’
        - Metrics = ‘accuracy’
    - Training configuration:
        - Epochs = 200
        - Batch size = 32
        - Validation split = 0.2 (20%)
- Test Accuracy: 80.08%
- Classification Report:
    - Class 0: Precision 0.75, Recall 0.87, F1-Score 0.81
    - Class 1: Precision 0.86, Recall 0.73, F1-Score 0.79
- Evaluation Graphs

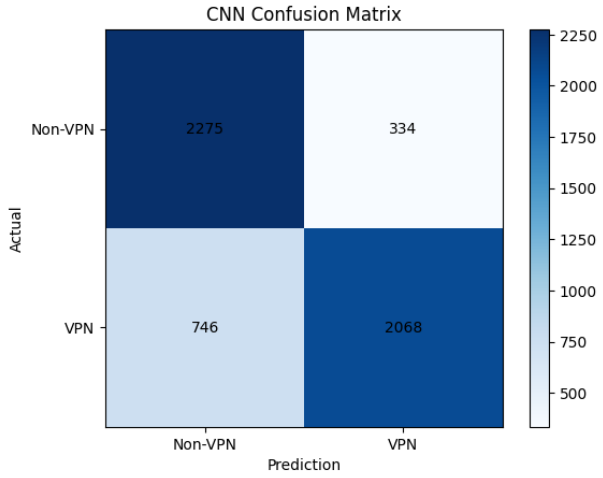

Fig. 21. CNN Confusion Matrix

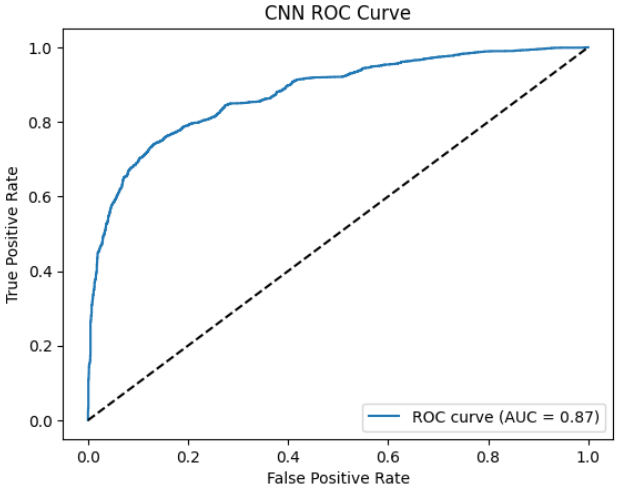

Fig. 22. CNN ROC Curve

5 ANALYSIS

This chapter presents the analysis of the results acquired in the previous chapter. Each model will be examined in detail and comparisons will be made between them.

5.1 Analysis 1: K-Nearest Neighbors (KNN)

In this experiment, KNN algorithm was applied over the dataset with k-values [3, 5, 7, 9, 11] and found k = 3 to be the most optimal as seen in the cross-validation graph (Fig. 4). The model achieved a decent test accuracy of 83.05%. The confusion matrix (Fig. 5) suggests that the misclassifications are relatively evenly distributed between FP and FN. The ROC curve (Fig. 6) also suggests that the distance-based matric used by KNN was effective at capturing patterns in the network traffic.

However, KNN's performance is inherently sensitive to scale distribution of input features. In datasets with high imbalanced data, this sensitivity might lead to reduced accuracy [Zhang, 2021]. The model also serves as a baseline for further tuning and optimization or needs additional preprocessing to increase performance in a real-time fraud detection system.

5.2 Analysis 2: Logistic Regression (LR)

In this experiment, LR algorithm was applied over the dataset with a range of regularization strengths C [0.01, 0.1, 1, 10, 100] and C = 100 was chosen as the best parameter through cross validation. Using this the model achieved a much lower test accuracy of 64.11% compared to KNN.

The reason for this is likely because LR inherently models a linear relationship on the logit scale. This indicates that the model's linear decision boundary may be too simplistic to capture the non-linear complexity of the dataset. Moreover, it can be sensitive to problems like multicollinearity and outliers which may further decrease performance [Fernandes et al., 2020].

5.3 Analysis 3: Support Vector Machines (SVM)

In this experiment, SVM algorithm was applied over the dataset with a radial basis function (RBF) kernel with parameters C = 100 and gamma = 1. This gave an average test accuracy of 78.14%. The cross-validation graph (Fig. 7) demonstrates these findings showing the highest performance for the said parameters.

The confusion matrix (Fig. 8) shows that the model classifies majority of the instances is not highly imbalanced between the classes. The ROC curve (Fig. 9) supports this analysis

by showing the tradeoff between the TPR and FPR.

SVM's performance can suffer when trained over large datasets due to increased computational costs and sensitivity to kernel parameters [Ahmad et al., 2018]. SVMs are usually robust when dealing with high dimensionality, non-linear or noisy data, but they require careful parameter tuning and optimization and may not scale as well when data complexity increases [Sha'abani et al., 2020].

5.4 Analysis 4: Naïve Bayes (NB)

NB achieved the lowest performance, having test accuracy of 54.23%. There is a massive imbalance between the precision and recall for class 0, where the accuracy was 0.82 and the recall was 0.06. Whereas for class 1 (VPN traffic), the recall was almost a perfect score of 0.99 but a lower precision of 0.53. The confusion matrix showed that for class 1, out of 2814 instances, only 37 were FP, however for class 0, out of 2609 instances, only 164 where TP. NB requires a very large number of instances to obtain good results and is usually less accurate compared to other classifiers on some datasets [Jadhav and Channe, 2016]. This reveals that the model overcompensates by classifying most instances as class 1. This discrepancy shows that the independence assumption of NB is too strong for this dataset.

5.5 Analysis 5: Random Forest (RF)

In this experiment, RF algorithm was applied over the dataset with parameters number of estimators = 100 and maximum depth 20 and achieved a test accuracy of 92.47%. RF is an ensemble method that builds multiple decision trees on bootstrapped subsets of the training data and then aggregates their predictions via majority voting [IBM, 2021].

The confusion matrix (Fig. 10) shows a balanced performance for both the classes, with a comparatively high precision and recall.

The ROC curve (Fig. 11) shows a high Area Under Curve (AUC), confirming the model's ability to capture patterns in the dataset.

These outcomes align with the literature which highlights ensemble methods like RF are well fitted in capturing non-linear features and handling high-dimensional data problems that simpler models often struggle with. This makes RF a good potential algorithm for real-time implementation in fraud detection systems.

5.6 Analysis 6: Gradient Boosted Trees (GBT)

In this experiment, GBT algorithm was applied over the dataset with parameters number of estimators = 200, learning rate = 20% and maximum depth 7 and achieved the highest test accuracy in this study of 92.60%. The cross-validation graph (Fig. 12) confirms these best parameters. GBT builds an ensemble of decision trees sequentially, where each new tree is trained to correct the errors (residuals) of the previous ensemble using gradient descent [Gaurav, 2021].

The confusion matrix (Fig. 13) shows that the model achieved a high precision and recall for both classes and is balanced. The ROC curve (Fig, 14) shows a high AUC.

This confirms that it can capture complex, non-linear patterns in the dataset and the results match with the literature. This makes GBT a good potential algorithm for real-time implementation in fraud detection systems.

5.7 Analysis 7: Decision Trees (DT)

In this experiment, DT algorithm was applied over the dataset and achieved a solid test accuracy of 89.39%. The parameters chosen were max_depth = 20, min_samples_split = 2, min_samples_leaf = 1.

The confusion matrix (Fig. 15) shows a balanced classification for the classes, although there were a few misclassifications suggesting the model may overfit features with noisy or ambiguous data. The ROC curve (Fig. 16) confirms a relatively high AUC.

Overall, the model can capture non-linear relationships in the dataset. However, it does not reach the level of performance reached by ensemble methods, suggesting that the simplicity of a single tree limits the model's ability to generalize new and unseen data

5.8 Analysis 8: Multilayer Perceptron (MLP)

In this experiment, MLP algorithm was applied over the dataset and achieved a test accuracy of 72.98%. The parameters chosen were hidden layers of sizes = (100, 50), ReLU activation, and the Adam solver.

The confusion matrix (Fig. 17) shows that while the model can classify some features of the dataset, however, it misclassifies a notable number of instances. Furthermore, the ROC curve (Fig. 18) shows an AUC in the moderate range.

These findings show a common issue with neural network models, like the need for extensive hyperparameter tuning and optimization, possible overfitting when the network is too deep or not properly regularized, and the requirement for a large amount of training data.

5.9 Analysis 9: Recurrent Neural Network (RNN)

In this experiment, the RNN algorithm was applied over the dataset and achieved a test accuracy of 74.34%. The parameters were SimpleRNN layer with 32 units (using tanh activation), a dense layer with 16 neurons (ReLU activation), and an output layer with a single neuron using sigmoid activation for binary classification.

The confusion matrix (Fig. 19) shows that the model can classify both classes with a relatively balanced precision and recall, although there is a notable number of misclassifications. The ROC curve (Fig. 20) shows a moderate AUC showing that while decent, it's not in par with the ensemble models. RNNs are primarily designed for

sequential data and might not be able to capture the complex non-linear patterns of the dataset and hence may require additional preprocessing or hyperparameter optimization.

5.10 Analysis 10: Convolutional Neural Network (CNN)

In this experiment, the CNN algorithm was applied over the dataset and achieved a test accuracy of 80.08%. The architecture consists of multiple 1D convolutional layers that extract local feature patterns from the data, max pooling layers to reduce dimensionality and control overfitting. The output from these layers was then passed through one or more fully connected layers with a sigmoid activation function to achieve a binary classification.

The confusion matrix (Fig. 21) shows that the model achieves a relatively balanced performance, however there are a notable number of misclassifications. The ROC curve (Fig. 22) shows a moderate AUC.

The results show that there may be a need for further hyperparameter tuning or additional regularization techniques.

5.11 Performance Summary

Table. 4. Model performance summary

Model	Test Accuracy
KNN	83.05%
LR	64.11%
SVM	78.14%
NB	54.23%
RF	92.47%
GBT	92.60%
DT	89.39%
MLP	72.98%
RNN	74.34%

CNN	80.08%

5.12 Summary

The results indicate that ensemble-based and tree-based models, that are GBT, RF and DT, perform better than other models in capturing the complicated decision boundaries in the dataset. This is likely because of their inherent ability to handle non-linearity. The performance received aligns with the trends in the literature, which suggests that ensemble methods tend to offer improved robustness and accuracy for VPN classification tasks.

On the other hand, simpler models such as LR and NB struggle with the datasets complexity which also aligns with the trends in literature. This is likely due to the linear assumptions in LR and feature independence assumptions in NB, as evidenced by the high imbalance of precision and recall values.

The moderate performance of the other models (CNN, KNN, etc.) suggests that while they can extract and model underlying patterns, they may require further tuning or additional data preprocessing to reach levels achieved by ensemble techniques.

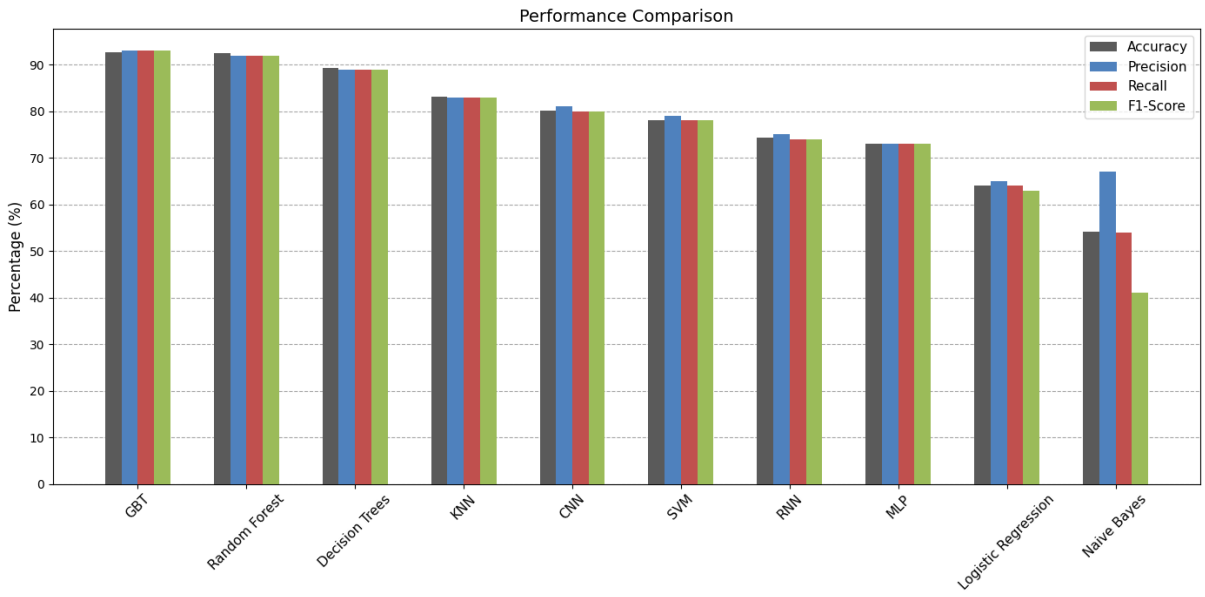

Fig. 23. Model performance comparison

6 DISCUSSION

This chapter contains the implications of the findings obtained in the previous chapters, this is situating the results within a wider context of fraud detection and VPN analysis as described in the literature. This chapter aims to connect the methodology, results and analysis with established theories and current emerging trends in the industry.

6.1 Integration with existing literature

The superior performance of ensemble-based models, especially RF and GBT, aligns with studies which reported high test accuracies and ROC values. This study extends prior work by systematically comparing a wider range of AI models, demonstrating that simple models like LR and NB struggle with complex non-linear patterns present in VPN traffic [Al-Fayoumi, Al-Fawa'reh, and Nashwan, 2022; Bagui et al., 2017].

Furthermore, the paper explores neural architectures (MLP, RNN, and CNN) which on its own does not provide high accuracies but with better optimization, parameters, preprocessing and overfitting prevention, shows promise for better performance as a hybrid model rather than standalone models [Lenka and Tiwari, 2020; Dutta and Bandyopadhyay, 2020].

6.2 Methodology

The methodological framework in this research is based on early pattern discovery techniques and integrates hyperparameter optimization to address data imbalance and overfitting [Gabbur et al., 2011; Kelly, 2019; Al-Fayoumi, Al-Fawa'reh, and Nashwan, 2022].

The evaluation strategies used are widely known and industry standard, including precision, recall, F1-score and accuracy [Chugh, Malik, Gupta, and Alkahtani, 2025].

6.3 Real world implications

The findings from this study are particularly relevant for the security of digital payment systems, specifically when there is use of VPNs to bypass location-based authentication. Institutions can leverage the high accuracy and performance of ensemble models to improve real-time VPN based fraud detection. However, this study also highlights the challenge of scalability and computational efficiency when deploying such complex models in large volume, real-world scenarios.

7 CONCLUSION

This study started to address the growing threat of VPN-based fraud in digital payment systems by exploring and comparing 10 AI models using network analysis of VPN encrypted traffic. This fills in and builds upon the gap in literature specific to this topic and provides insight on ensemble models and hybrid techniques for real-time VPN-based fraud detection and VPN detection in general.

7.1 Motivation and Goals

The motivation for this research stemmed from the increased use of VPNs by fraudsters to bypass geo-location-based authentication systems that are critical to ensure regulatory compliance and customer trust. The main objectives of this study were:

- Developing and implementing a wide range of ML models (namely KNN, LR, SVM, NB, DT, RF, GBT, MLP, RNN, and CNN) for detecting VPN encrypted traffic.
- Employing network analysis techniques to capture features of the VPN encrypted traffic dataset.
- Evaluating and comparing the performance of these models using industry standard classification metrics and identify the best approach for real-world scenarios.

7.2 Contributions

- **Comprehensive model comparison**: This study provides a systematic comparison of AI models' performances and indicating models best in capturing non-linear patterns in VPN encrypted traffic.

- **Integration of network analysis with machine learning**: This study builds on existing literature including network features analysis, including packet lengths, flow durations, timeouts, and other calculated metrics.

- **Hyperparameter optimization**: This study also highlights the need for parameter tuning for model optimization for getting the best performances. The use of grid search and cross-validation methods are necessary in minimum error situations of fraud detection.

- **Real-world deployment**: This study provides practical insights into scalability and efficiency different AI models in detecting VPN encrypted traffic.

7.3 Limitations and Future Work

While this research provides evidence and explanations for the effectiveness of ensemble and hybrid models in VPN detection, some limitations remain:

- **Scalability and Efficiency**: The computational demands for models like GBT and RF in real time applications are not insignificant. There needs to be work done on optimization techniques and hardware acceleration for such models to be deployed in real-world applications.
- **Data manipulation**: The performance disparity seen between neural network models shows that there need to be specific improvements in data preprocessing and feature selection to improve their performance.
- **Integration of network analysis**: Despite the results of this study, the live integration of network analysis with ML remains underdeveloped. Further studies are needed to create holistic models that leverage both statistical and network-based insights.

7.4 Problems faced

The original plan of this dissertation was to simulate VPN traffic using GNS3, sockets and virtual machines. This was discontinued due to technical issues.
ISCXVPN2016 dataset contains the exact same data that was needed for the study which included VPN and non-VPN encrypted traffic over multiple protocols.

REFERENCES

www.unb.ca. (n.d.). VPN 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. [online] Available at: https://www.unb.ca/cic/datasets/vpn.html.

Cybernews. (2022). Save on YouTube Premium Individual & Family Plans with a VPN! [online] Available at: https://cybernews.com/how-to-use-vpn/youtube-premium-discount/ [Accessed 19 Mar. 2025].

van Hardeveld, G.J., Webber, C. and O'Hara, K. (2016). Discovering credit card fraud methods in online tutorials. Proceedings of the 1st International Workshop on Online Safety, Trust and Fraud Prevention - OnSt '16. doi:https://doi.org/10.1145/2915368.2915369.

Prasad Gabbur, Sharath Pankanti, Fan, Q. and Trinh, H. (2011). A pattern discovery approach to retail fraud detection. doi:https://doi.org/10.1145/2020408.2020460.

Kelly, C. (2019). EXPERIENTIAL METHODS FOR IDENTIFYING AND REDUCING POINT OF SALE RETAIL FRAUD. EDPACS, 59(5), pp.13–20. doi:https://doi.org/10.1080/07366981.2019.1603834.

Sahin, Y. & Duman, E. (2011). Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 (IMECS 2011), March 16-18, Hong Kong, pp. 442-447. ISBN: 978-988-18210-3-4. [accessed 19 October, 2024]

Chugh, B., Malik, N., Gupta, D. and Alkahtani, B.S. (2025). A probabilistic approach driven credit card anomaly detection with CBLOF and isolation forest models.

Taniguchi, M., Haft, M., Jaakko Hollmén and Volker Tresp (2002). Fraud detection in communication networks using neural and probabilistic methods. doi:https://doi.org/10.1109/icassp.1998.675496.

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M. and Anderla, A. (2019). Credit Card Fraud Detection - Machine Learning methods. *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*. [online] doi:https://doi.org/10.1109/infoteh.2019.8717766.

Awoyemi, J.O., Adetunmbi, A.O. and Oluwadare, S.A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI), [online] pp.1–9. doi:https://doi.org/10.1109/iccni.2017.8123782.

Dornadula, V.N. and Geetha, S. (2019). Credit Card Fraud Detection using Machine Learning Algorithms. Procedia Computer Science, [online] 165, pp.631–641. doi:https://doi.org/10.1016/j.procs.2020.01.057.

Zain ul Abideen, M., Saleem, S. and Ejaz, M. (2019). VPN Traffic Detection in SSL-Protected Channel. Security and Communication Networks, 2019, pp.1–17. doi:https://doi.org/10.1155/2019/7924690.

Hines, C. and Youssef, A. (2019). Class Balancing for Fraud Detection in Point Of Sale Systems. 2021 IEEE International Conference on Big Data (Big Data), pp.4730–4739. doi:https://doi.org/10.1109/bigdata47090.2019.9006040.

Al-Fayoumi, M., Al-Fawa'reh, M. and Nashwan, S. (2022). VPN and Non-VPN Network Traffic Classification Using Time-Related Features.

Alamleh, A., Nasir, O., Hannoun, A.A., Darwish, M., Alallawi, B., Alshurafa, Z. and Khouj, M. (2024). Real-Time VPN Anomaly Detection System. *2024 25th International Arab Conference on Information Technology (ACIT)*, [online] pp.1–7. doi:https://doi.org/10.1109/acit62805.2024.10877255.

Sajid, R.Y. and Pekar, A. (2025). *Binary VPN Traffic Detection Using Wavelet Features and Machine Learning*. [online] arXiv.org. Available at: https://arxiv.org/abs/2502.13804 [Accessed 24 Mar. 2025].

Goel, A., Kashyap, A., Reddy, B.D., Kaushik, R., Nagasundari, S. and Honnavali, P.B. (2022). Detection of VPN Network Traffic. *2022 IEEE Delhi Section Conference (DELCON)*. doi:https://doi.org/10.1109/delcon54057.2022.9753621.

Lenka, S. and Tiwari, R. (2020). Real-Time Fraud Prevention in Digital Wallet Transactions Using CNN-RNN Hybrid Networks.

Dutta, S. and Bandyopadhyay, S.K. (2020). Detection of Fraud Transactions Using Recurrent Neural

Network during COVID-19.

Bagui, S., Fang, X., Kalaimannan, E., Bagui, S.C. and Sheehan, J. (2017). Comparison of machine-learning algorithms for classification of VPN network traffic flow using time-related features. Journal of Cyber Security Technology, 1(2), pp.108–126. doi:https://doi.org/10.1080/23742917.2017.1321891.

Zhang, S. (2021). Challenges in KNN Classification. *IEEE Transactions on Knowledge and Data Engineering*, 34(10), pp.1–1. doi:https://doi.org/10.1109/tkde.2021.3049250.

Ahmad, I., Basheri, M., Iqbal, M.J. and Rahim, A. (2018). Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection. *IEEE Access*, 6, pp.33789–33795. doi:https://doi.org/10.1109/access.2018.2841987.

Sha'abani, M.N.A.H., Fuad, N., Jamal, N. and Ismail, M.F. (2020). kNN and SVM Classification for EEG: A Review. *Lecture Notes in Electrical Engineering*, pp.555–565. doi:https://doi.org/10.1007/978-981-15-2317-5_47.

Jadhav, S.D. and Channe, H.P. (2016). Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques. *International Journal of Science and Research (IJSR)*, 5(1), pp.1842–1845. doi:https://doi.org/10.21275/v5i1.nov153131.

IBM (2021). *Random Forest*. [online] Ibm.com. Available at: https://www.ibm.com/think/topics/random-forest.

Gaurav (2021). *An Introduction to Gradient Boosting Decision Trees - Machine Learning Plus*. [online] Machine Learning Plus. Available at: https://www.machinelearningplus.com/machine-learning/an-introduction-to-gradient-boosting-decision-trees/#Ensemble-Learning [Accessed 23 Mar. 2025].

A APPENDIX: PLES (Professional, Legal, Ethical, and Social Issues)

A.1 Professional Issues:

Developing an AI-based fraud detection system for VPN-based fraud in digital payment systems aligns with professional standards in software development and cybersecurity. The proposed system adheres to the BCS Code of Conduct, ensuring integrity, public interest and a focus on high-quality solutions. The design will follow modular principles for scalability and maintenance and will address industry standards in terms of quality.

A.2 Legal Issues:

The system should comply with data protection standards such as General Data Protection Regulation (GDPR) to safeguard any collected VPN traffic data. All data handling processes will ensure privacy and confidentiality. Adherence to computer misuse laws is important to note when simulating VPN based fraud scenarios to prevent unintended violations.

A.3 Ethical Issues:

Ethical considerations are a vital consideration to ensure responsible use of AI. There must be transparency in algorithm design and clear explanation of decision-making processes to avoid any biases. The model must be able to minimize false positives, reducing the risk of financial harm to users or merchants. Since this model will be trained on anonymized VPN traffic data, there is no infringement of individual privacy.

A.4 Social Issues:

VPN-based digital payments fraud in general undermines trust in digital payment systems. By addressing VPN based fraud, this project contributes to the security and reliability of financial transactions, building greater public trust in digital payment systems. Another highlight is the importance of collaboration between financial institutions and regulatory authorities to combat such sophisticated fraud techniques.